



Office de la propriété
intellectuelle
du Canada

Un organisme
d'Industrie Canada

Canadian
Intellectual Property
Office

An Agency of
Industry Canada

1033 U.S. PTO
09/872827
05/31/01

Bureau canadien
des brevets
Certification

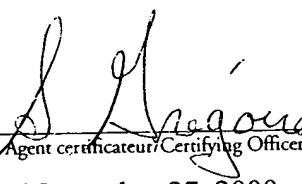
La présente atteste que les documents
ci-joints, dont la liste figure ci-dessous,
sont des copies authentiques des docu-
ments déposés au Bureau des brevets.

Canadian Patent
Office
Certification

This is to certify that the documents
attached hereto and identified below are
true copies of the documents on file in
the Patent Office.

Specification and Drawings, as originally filed, with Application for Patent Serial No:
2,322,603, on October 6, 2000, by **IBM CANADA LIMITED - IBM CANADA**
LIMITÉE, assignee of Iqbal A. Goralwalla, Michael J. Winer, David C. Sharpe and
Matthew A. Huras, for "Optimizing Updatable Scrollable Cursors in Database System".

**CERTIFIED COPY OF
PRIORITY DOCUMENT**


Agent certificateur/Certifying Officer

November 27, 2000

Date

Canada

(CIPO 68)

O P I C  C I P O

OPTIMIZING UPDATABLE SCROLLABLE CURSORS
IN DATABASE SYSTEMS

ABSTRACT

5 An optimization for updatable scrollable cursors in database systems. To ensure that a temporary copy of a data record matches the data record maintained in the database, the temporary copy is associated with a timestamp included on each page storing the data records of the database. The timestamp on a page in the database is updated whenever any record stored on the page is updated.

10 The optimization compares the timestamp stored with the temporary record with the current timestamp associated with the page on which the data record is stored in the database. If the two timestamps are matching then the temporary record is identical to the database table record and there is no need to carry out a comparison of the attribute values of the temporary record with the record in the database table.

OPTIMIZING UPDATABLE SCROLLABLE CURSORS
IN DATABASE SYSTEMS

5 **FIELD OF THE INVENTION**

The present invention is directed to an improvement in computing systems and in particular to optimizing command execution in computer database systems that provide for updatable scrollable cursors.

10 **BACKGROUND OF THE INVENTION**

The Open Database Connectivity (ODBC) specification supports updatable scrollable cursors for Relational Database Management Systems (RDBMSs). This standard provides that multiple cursors may be defined for tables in relational databases and that positioned UPDATEs and DELETEs may be performed on the tables based on the scrollable cursor's location. The ODBC standard also provides for an attribute in the database to define an optimistic concurrency scheme. In the ODBC standard this attribute is referred to as SQL_CONCUR_VALUES. Use of the SQL_CONCUR_VALUES attribute provides that a positioned UPDATE or DELETE succeeds only if the record data to be modified has not been changed since it was last fetched by the user.

20 In certain relational database systems such as the DB2 UDB (trade-mark) RDBMS, for each scrollable cursor a temporary copy of record data is made when the data is fetched by the user. Where the SQL_CONCUR_VALUES attribute is applied, and a positioned UPDATE or DELETE is to be carried out, the temporary copy of record data must be compared to the current record data in the database to ensure that the record data has not changed since the time that it was copied to the temporary location. A comparison of the record in the temporary copy with the record in the current table may result in significant overhead cost for the UPDATE or DELETE where the records to compare are extensive.

25 It is therefore desirable to have a relational database system that will support the ODBC updatable

scrollable cursors and the SQL_CONCUR_VALUES attribute in which it is possible to optimize the steps to carry out the positioned UPDATE or DELETE commands.

SUMMARY OF THE INVENTION

5 According to one aspect of the present invention, there is provided an improved system for optimizing updatable scrollable cursors in database systems.

According to another aspect of the present invention, there is provided a method for optimizing command execution in a database system, the database system storing data records on data pages,
10 a log sequence number being maintained in association with each data page, the log sequence number including a time stamp indicating the time of the last modification of data on the data page, the database system supporting the selective copying of a source data record from a specified data page into a temporary data record in a temporary data structure, the method including the steps of

15 storing a reference log sequence number in association with a temporary data record on the selective copying of a source data record to the temporary data record, the reference log sequence number representing the log sequence number of the specified data page at the time the source data record is copied to the temporary data record, and

20 determining that the source data record remains unmodified since the time that the source data record is copied to the temporary data record by comparing the reference log sequence number for the temporary data record with the current log sequence number of the specified data page.

According to another aspect of the present invention, there is provided a method for optimizing positioned UPDATE and DELETE command execution in a relational database system supporting scrollable cursors and optimistic concurrency, the database system storing data records on data
25 pages, a log sequence number being maintained in association with each data page, the log sequence number including a time stamp indicating the time of the last modification of data on the data page,

the database system fetching data in response to a user request by selectively copying a source data record from a specified data page into a temporary data record in a temporary data structure, the method including the steps of:

5 in conjunction with the selective copying of data into a temporary data record, the step of storing a reference log sequence number in association with the temporary data record, the reference log sequence number representing the log sequence number of the specified data page at the time the source data record is copied to the temporary data record,

10 in response to a request for execution of a positioned UPDATE or DELETE command, determining that the source data record remains unmodified since the time that the source data record is copied to the temporary data record by comparing the reference log sequence number for the temporary data record with the current log sequence number of the specified data page, and

15 where the source data record remains unmodified, carrying out the step of executing the UPDATE or DELETE command without comparing values of the attributes in the temporary data record and the source data record.

20 According to another aspect of the present invention, there is provided a computer program product for a database management system, the computer program product including a computer usable medium having computer readable code means embodied in said medium, including computer readable program code means for carrying out the above methods.

25 According to another aspect of the present invention, there is provided a relational database management system including, data records stored on data pages, each data page including a log sequence number, the log sequence number for a page being updated on a modification being made to a data record stored on the said page, means for copying a specified data record from the data page containing the data record to a temporary data record in a temporary table, means for associating the value of the log sequence number of the data page containing the data record with the temporary data

record, at the time of copying the data record into the temporary data record, means for comparing the associated log sequence number of the temporary data record with a current log sequence number of the data page containing the data record to determine if a comparison of attribute values of the data record and of the temporary data record is required to verify that the records match each other.

5

According to another aspect of the present invention, there is provided a relational database system supporting positioned UPDATE and DELETE command execution, scrollable cursors and optimistic concurrency, the relational database system storing data records on data pages, a log sequence number being maintained in association with each data page, the log sequence number including a time stamp indicating the time of the last modification of data on the data page, the relational database system fetching data in response to a user request by selectively copying a source data record from a specified data page into a temporary data record in a temporary data structure, the relational database system including means for storing a reference log sequence number in association with the temporary data record in conjunction with the selective copying of data into a temporary data record, the reference log sequence number representing the log sequence number of the specified data page at the time the source data record is copied to the temporary data record, means for determining that the source data record remains unmodified since the time that the source data record is copied to the temporary data record, in response to a request for execution of a positioned UPDATE or DELETE command, by comparing the reference log sequence number for the temporary data record with the current log sequence number of the specified data page, and means for carrying out the step of executing the UPDATE or DELETE command without comparing values of the attributes in the temporary data record and the source data record., where the source data record is determined to have remained unmodified.

20

Advantages of the present invention include a reduction in processing time of database UPDATE or DELETE operations based on scrollable cursors where there is support for optimistic concurrency.

25

BRIEF DESCRIPTION OF THE DRAWINGS

The preferred embodiment of the invention is shown in the drawings, wherein:

Figure 1 is a block diagram representing example tables in a database subject to the optimization of the preferred embodiment.

In the drawings, the preferred embodiment of the invention is illustrated by way of example. It is
5 to be expressly understood that the description and drawings are only for the purpose of illustration and as an aid to understanding, and are not intended as a definition of the limits of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 illustrates, in a block diagram, data which is subject to the optimization of the preferred
10 embodiment. Figure 1 shows a portion of a relational database containing rows m and n, shown as records 12, 14 in the figure. Figure 1 also shows a temp table 16 in which data corresponding to rows m and n are shown as records 18, 20.

In the preferred embodiment, records in the relational database are stored on pages. Each page has
15 a log sequence number (LSN) associated with the page. In Figure 1, data page 10 is shown with associated LSN 22. The LSN for the data page includes information which effectively provides a time stamp of the last modification made to any table data (records) on that page. With reference to the example of Figure 1, LSN 22 is updated when either record 12 or record 14, or any other record stored on data page 10, is modified in any way.

20 As is indicated in the diagram of Figure 1, according to the preferred embodiment, a copy of the data page LSN is maintained in association with a record when that record is written to temp table 16. Temp table 16 is used to copy row values when an updatable scrollable cursor is used to retrieve table record values for a user. In the example of Figure 1, a cursor has been used to access rows m
25 and n and therefore the row m value in record 12 in data page 10 is copied to record 18 in temp table 16. Similarly the row n value in record 14 is copied to temp table 16 record 20 when a cursor in table is used to fetch row n for a user.

In the preferred embodiment, when record 18 having the value of row m is stored in temp table 16,

a copy of LSN 22 is made and stored in the temp table in association with record 18. This is shown in Figure 1 as LSN 24. Similarly, a copy of LSN 22 is made in association with record 20 when the value of row n is copied to temp table 16. This associated LSN value is shown as LSN 26 in Figure 1.

5

Due to the concurrency available in the ODBC standard, it is possible for row m to be copied to temp table 16 from data page 10 with the then current value of a LSN 22 being copied to LSN 24 and to then have a subsequent modification to data page 10 before the cursor reaches row n (in the table record 14). As a result, row n values may be copied into record 20 in temp table 16 with LSN value 10 26 that differs from LSN 24. This is due to a change to the value of LSN 22 when the data page 10 values are modified prior to copying the value of row n into temp table 16.

In the preferred embodiment, where SQL_CONCUR_VALUES attribute is associated with the table containing data page 10, and an UPDATE or DELETE operation is specified for, for example, row 15 m or row n, based on the position of a cursor, it is necessary to ensure that the value of rows m and n as stored in temp table 16 are the same as the values as stored in data page 10. Although this may be carried out by a direct comparison of the attribute values in the respective rows in temp table 16 and data page 10, the preferred embodiment is able to potentially avoid such a direct comparison by comparing the LSN values of the records. Where, for example, row m is subject to a positioned 20 UPDATE or DELETE based on a scrollable cursor, the value of LSN 24 is compared with the value of LSN 22. Where these values match, it is necessarily the case that record 12 has not been modified since a copy of the value of row m was copied into record 18 in temp table 16. If record 12 had been modified, the value of LSN 22 would have changed and there would therefore not be a match between the value of LSN 22 and the value of LSN 24. In this manner, it is possible to avoid 25 comparing all attribute values of record 12 and of record 18 by carrying out the simple comparison of the value of LSN 24 and the value of LSN 22. Where these values are different, it will be necessary to carry out the direct comparison of record 12 and record 18. However, this step may be avoided where the values of LSN 24 and LSN 22 are found to match. Where this is the case, the user will be able to directly update the value of record 12 and rely on the record 18 value as

corresponding to record 12.

In the implementation of the preferred embodiment, the retrieval and comparison of LSN values is carried out using an internal command referred to as FETCH SENSITIVE NO DATA. The operations carried out by the FETCH SENSITIVE NO DATA internal command result in the LSN for the data page and the LSN for the record in the temp table being retrieved and compared, as is described above. In this way, the RDBMS of the preferred embodiment is able to execute the FETCH SENSITIVE NO DATA command as a part of the steps taken by the RDBMS in optimizing execution of UPDATE or DELETE commands in response to a user request.

10

The preferred embodiment is described above with respect to the UPDATE and DELETE commands where a RDBMS supports optimistic concurrency for a scrollable cursor. The optimization of the preferred embodiment may also be used in implementing other commands in an RDBMS which include a temporary table copy of a record, and require a confirmation that the temp table copy is equivalent to the database copy.

15

20

25

An example of such an implementation of the optimization of the preferred embodiment is with respect to the FETCH SENSITIVE command in the DB2 UDB RDBMS. The FETCH SENSITIVE command is available to users (in contrast to the FETCH SENSITIVE NO DATA command referred to above which is used internally in the RDBMS, only). Execution of the command, without any optimization, results in the fetch of a record from the database table and the qualification of that record (its attribute values are compared with the SQL predicates associated with the command). Where the record qualifies, the temp table is updated and the record is returned to the user. The optimization of the preferred embodiment makes it possible to use the value of the LSN stored in the temp table to avoid steps in carrying out the command. Where the row m, for example, has been previously fetched and is in temp table 16, and the value of LSN 24 is equivalent to data page 10 LSN 22, a FETCH SENSITIVE command carried out on row m may be implemented by positioning the cursor at the appropriate record and returning a flag to the user to indicate that the previously fetched values remain current. If the two LSN values are not equal, then the non-optimized steps

to carryout the FETCH SENSITIVE command are followed. Using the comparison of the LSN value associated with the temp table record and the LSN value of the data page, the copying of attribute values to the temp table may be avoided. In this manner, the optimization of the preferred embodiment may be used to increase efficiency in carrying out command execution in a database
5 with updatable scrollable cursors.

Although a preferred embodiment of the present invention has been described here in detail, it will be appreciated by those skilled in the art, that variations may be made thereto, without departing from the spirit of the invention or the scope of the appended claims.

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A method for optimizing command execution in a database system, the database system storing data records on data pages, a log sequence number being maintained in association with each data page, the log sequence number comprising a time stamp indicating the time of the last modification of data on the data page, the database system supporting the selective copying of a source data record from a specified data page into a temporary data record in a temporary data structure, the method comprising the steps of:

5 a. storing a reference log sequence number in association with a temporary data record on the selective copying of a source data record to the temporary data record, the reference log sequence number representing the log sequence number of the specified data page at the time the source data record is copied to the temporary data record, and

10 b. determining that the source data record remains unmodified since the time that the source data record is copied to the temporary data record by comparing the reference log sequence number for the temporary data record with the current log sequence number of the specified data page.

20 2. A method for optimizing positioned UPDATE and DELETE command execution in a relational database system supporting scrollable cursors and optimistic concurrency, the database system storing data records on data pages, a log sequence number being maintained in association with each data page, the log sequence number comprising a time stamp indicating the time of the last modification of data on the data page, the database system fetching data in response to a user request by selectively copying a source data record from a specified data page into a temporary data record in a temporary data structure, the method comprising the steps of:

25 a. in conjunction with the selective copying of data into a temporary data record, the step of storing a reference log sequence number in association with the temporary data record, the reference log sequence number representing the log sequence number

of the specified data page at the time the source data record is copied to the temporary data record,

- 5 b. in response to a request for execution of a positioned UPDATE or DELETE command, determining that the source data record remains unmodified since the time that the source data record is copied to the temporary data record by comparing the reference log sequence number for the temporary data record with the current log sequence number of the specified data page, and
- 10 c. where the source data record remains unmodified, carrying out the step of executing the UPDATE or DELETE command without comparing values of the attributes in the temporary data record and the source data record.

15 3. A computer program product for a database management system, the computer program product comprising a computer usable medium having computer readable code means embodied in said medium, comprising computer readable program code means for carrying out the method of claim 1 or claim 2.

20 4. A relational database management system comprising:

data records stored on data pages, each data page comprising a log sequence number, the log sequence number for a page being updated on a modification being made to a data record stored on the said page,

means for copying a specified data record from the data page containing the data record to a temporary data record in a temporary table,

25 means for associating the value of the log sequence number of the data page containing the data record with the temporary data record, at the time of copying the data record into the temporary data record,

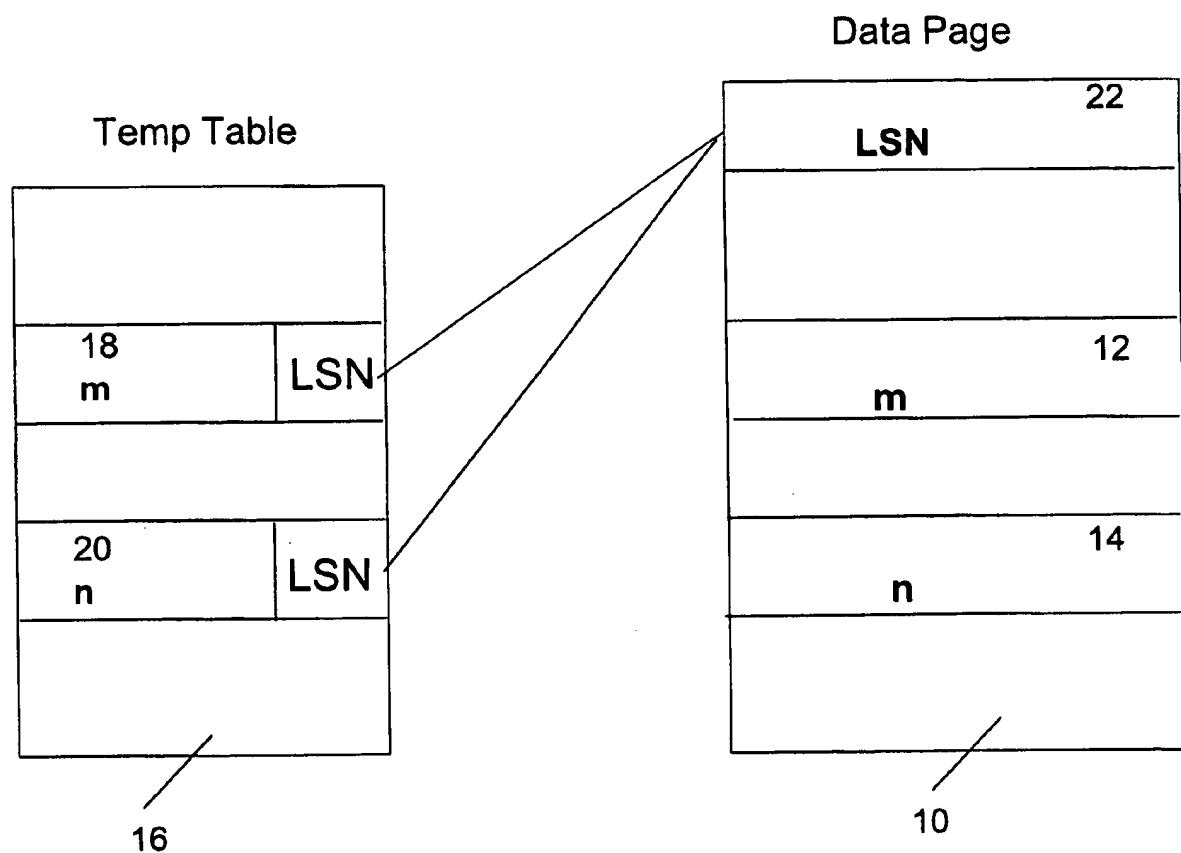
means for comparing the associated log sequence number of the temporary data record with a current log sequence number of the data page containing the data record to determine if a comparison of attribute values of the data record and of the temporary data record is required to verify that the records match each other.

5. A relational database system supporting positioned UPDATE and DELETE command execution, scrollable cursors and optimistic concurrency, the relational database system storing data records on data pages, a log sequence number being maintained in association with each data page, the log sequence number comprising a time stamp indicating the time of the last modification of data
5 on the data page, the relational database system fetching data in response to a user request by selectively copying a source data record from a specified data page into a temporary data record in a temporary data structure, the relational database system comprising

10 a. means for storing a reference log sequence number in association with the temporary data record in conjunction with the selective copying of data into a temporary data record, the reference log sequence number representing the log sequence number of the specified data page at the time the source data record is copied to the temporary data record,

15 b. means for determining that the source data record remains unmodified since the time that the source data record is copied to the temporary data record, in response to a request for execution of a positioned UPDATE or DELETE command, by comparing the reference log sequence number for the temporary data record with the current log sequence number of the specified data page, and

20 c. means for carrying out the step of executing the UPDATE or DELETE command without comparing values of the attributes in the temporary data record and the source data record., where the source data record is determined to have remained unmodified.

Fig. 1



Office de la propriété
intellectuelle
du Canada

Un organisme
d'Industrie Canada

Canadian
Intellectual Property
Office

An Agency of
Industry Canada

WPS

ROSEN, ARNOLD
IBM Canada Ltd.
Intellectual Property Department
Dept. B4/U59/3600/MKM
3600 Steeles Avenue East
MARKHAM Ontario
L3R 9Z7

Date : 2000/11/17

FILING CERTIFICATE

Application No.	: 2,322,603	Filing Date	: 2000/10/06
Expected Laid-Open Date	: 2002/04/06	Your Reference	: CA9-2000-0033
Title of Invention	: OPTIMIZING UPDATABLE SCROLLABLE CURSORS IN DATABASE SYSTEMS		
Applicant(s)	: IBM CANADA LIMITED-IBM CANADA LIMITEE		
Inventor(s)	: GORALWALLA, IQBAL A.; WINER, MICHAEL J.; SHARPE, DAVID C.; HURAS, MATTHEW A.		

The Request for Examination and prescribed fee have been made of record.

Special Notice

You are reminded that annual fees to maintain your application are needed for each one-year period between the 2nd and 20th anniversaries and must be paid on or before each anniversary. Failure to pay within the time limit will lead to abandonment of your application.

Commissioner of Patents

Canada

<http://opic.gc.ca> • Ottawa-Hull K1A 0C9 • <http://cipco.gc.ca>
OPIC-CIPO 191

O P I C C I P O